Sử dụng đoạn code như gợi ý

```
import pandas as pd
import numpy as np
blods=pd.read_csv("data.csv")
columns=list(blods.columns[1:-1])
blods.head()
```

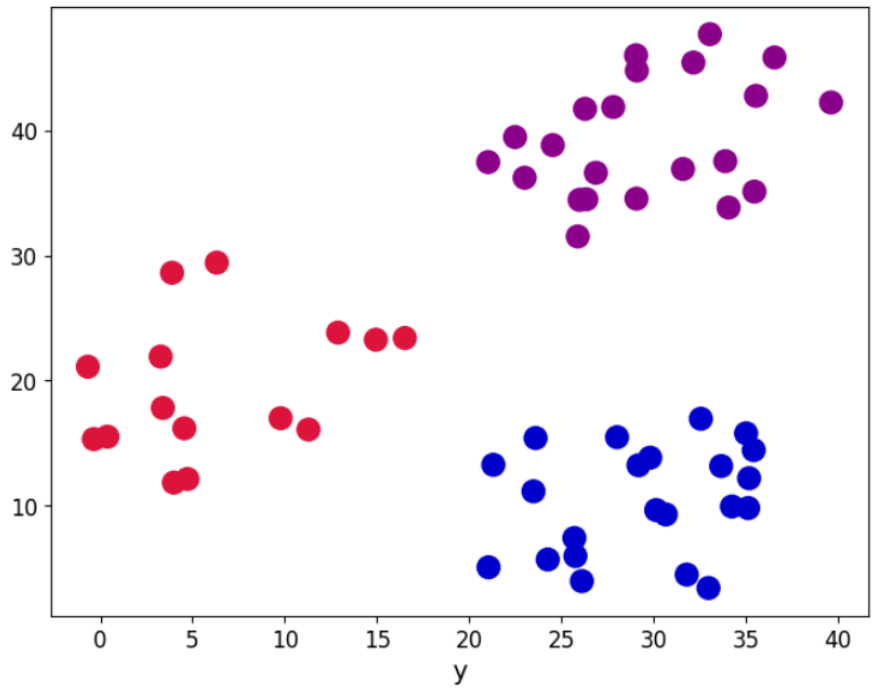
Dữ liệu được tạo theo yêu cầu

	ID	x	y	cluster
0	1	35.190	12.189	1
1	2	26.288	41.718	2
2	3	0.376	15.506	0
3	4	26.116	3.963	1
4	5	25.893	31.515	2

Vẽ phân phối các điểm dữ liệu

```
from matplotlib.colors import ListedColormap
import matplotlib.pyplot as plt

customcmap = ListedColormap(['crimson','mediumblue','darkmagenta'])
fig,ax = plt.subplots(figsize=(8,6))
plt.scatter(x=blods['x'],y=blods['y'],s=150,
            c=blods['cluster'].astype('category'),
            cmap=customcmap)
ax.set_xlabel(r'x',fontsize=14)
ax.set_xlabel(r'y',fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```

Trong đó
customcmap = ListedColormap(['crimson','mediumblue','darkmagenta'])
Với ListedColormap được sử dụng để tạo ra một colormap từ danh sách các màu được chỉ định.
c=blods['cluster'].astype('category') với cluster là xác loại màu được chỉ định



Chọn 3 điểm bất kỳ trong dataset để tạo thành 3 điểm trung tâm

```
#Khởi tạo tâm và xác định k
def initiable_centroids(k,dset):
    centroids = dset.sample(k)
    return centroids
np.random.seed(42)
k=3
df=blods[['x','y']]
centroids = initiable_centroids(k,df)
centroids
```

	x	y
0	35.190	12.189
5	23.606	15.402
34	23.492	11.142

Tính khoản cách

```
#tính khoảng cách
def rsserr(a,b):
    return np.square(np.sum((a-b)**2))
```

Tính toán RSS (Residual Sum of Squares) error là tổng bình phương khoảng cách
Không nên thay đổi đoạn code rsserr bằng các cách tính khác. Bởi vì sẽ làm ảnh hưởng đến điều kiện dừng khi xây dựng Kmedian và Kmeans

```
if j>0:
    if err[j-1]-err[j]<=tol:
        goahead =False
j+=1
```

với tol = 1e-4 = 0.0001

Điều chỉnh đoạn code so với mẫu để tính các giá tị trung tâm hợp lý

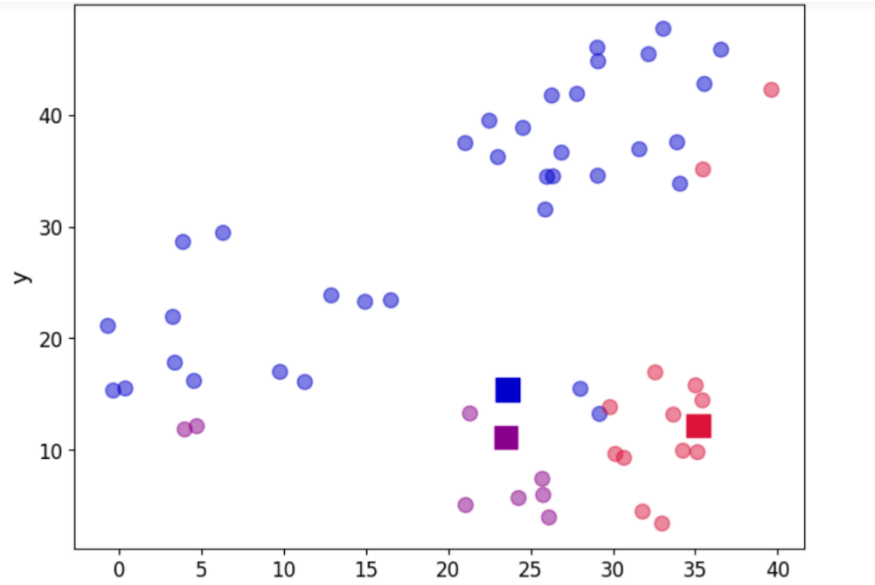
```
def centroid_assignation(dset, centroids):
    k = centroids.shape[0]
    n = dset.shape[0]
    assignation = []
    assign_errors = []
    for obs in range(n):
        all_errors = np.array([])
        for centroid in range(k):
            err = rsserr(centroids.iloc[centroid], dset.iloc[obs])
            all_errors = np.append(all_errors, err)
        nearest_centroid = np.argmin(all_errors)
        nearest_centroid_error = all_errors[nearest_centroid]
        assignation.append(nearest_centroid)
        assign_errors.append(nearest_centroid_error)
    return assignation, assign_errors
```

Xét từng điểm với các điểm trung tâm để tính toán khoảng cách. Tính khoản cách với rsserr và lưu lại với all_errors. Chọn các giá trị có giá trị nhỏ nhất đối với các giá trị được chọn làm trung tâm.

```
df['centroid'],df['error']=centroid_assignation(df,centroids)
df.head()
```

	x	y	centroid	error
0	35.190	12.189	0	0.000000
1	26.288	41.718	1	489615.047636
2	0.376	15.506	1	291215.340218
3	26.116	3.963	2	3413.295654
4	25.893	31.515	1	70150.362982

Trả về 2 giá trị gồm các centroid là các cụm của 3 điểm trung tâm và khoảng cách error



3 giá trị trung tâm và phân bổ màu với các điểm có khoảng cách gần nhất

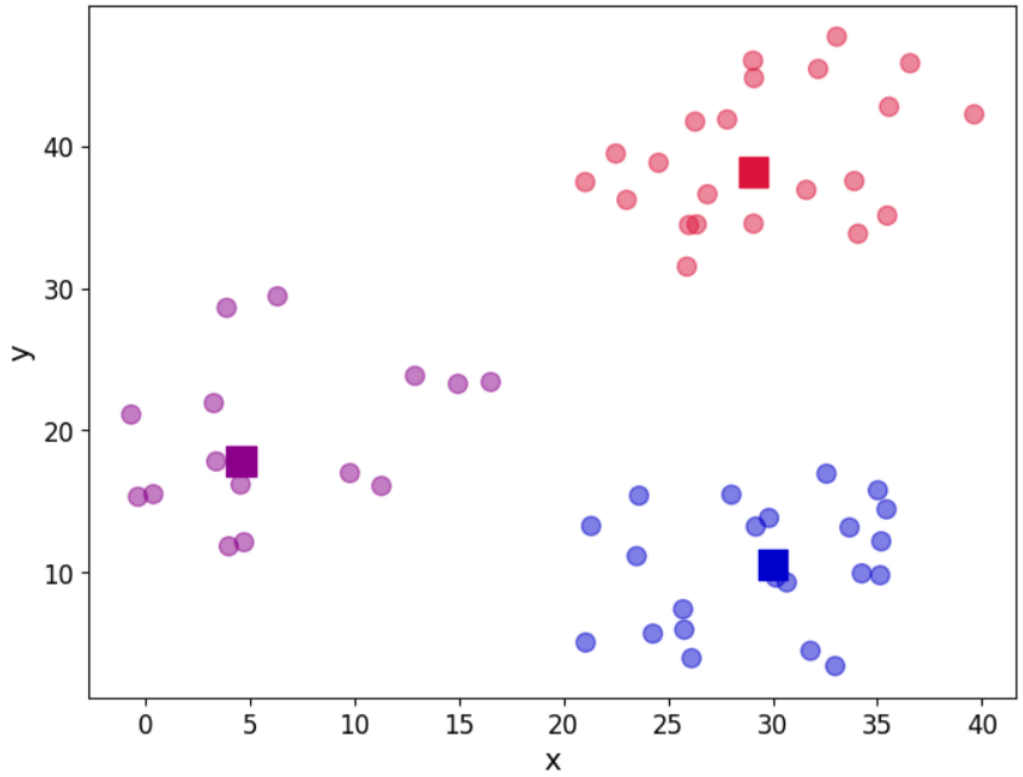
```
def kmedian(dset,k=2,tol=1e-4):
    working_dset = dset.copy()
    err=[]
    goahead = True
    j=0
    centroids = initiable_centroids(k,dset)
    while(goahead):
        working_dset['centroid'],j_err = centroid_assignation(working_dset,centroids)
        err.append(sum(j_err))
        centroids = working_dset.groupby('centroid').median().reset_index(drop=True)
        if j>0:
            if err[j-1]-err[j]<=tol:
                goahead =False
        j+=1

    working_dset['centroid'],j_egg = centroid_assignation(working_dset,centroids)
    centroids = working_dset.groupby('centroid').median().reset_index(drop=True)
    return working_dset['centroid'], j_err, centroids
```

```
df['centroid'],df['error'],centroids=kmedian(df[['x','y']],3)
df.head()
```

Xây dựng kmedian bằng cách tính khoản cách của tất cả các đuêmr so với 3 điểm trung tâm theo median. Điều kiện dừng xảy ra khi hiệu 2 tổng err vị trí trước lớn hơn 1 khoảng 0.0001 (giá trị khoảng cách không thay đổi được nữa)

	x	y	centroid	error
0	35.190	12.189	1	888.777539
1	26.288	41.718	0	410.436826
2	0.376	15.506	2	516.684359
3	26.116	3.963	1	3385.820331
4	25.893	31.515	0	2968.727738

Kết quả khi chạy Kmedian

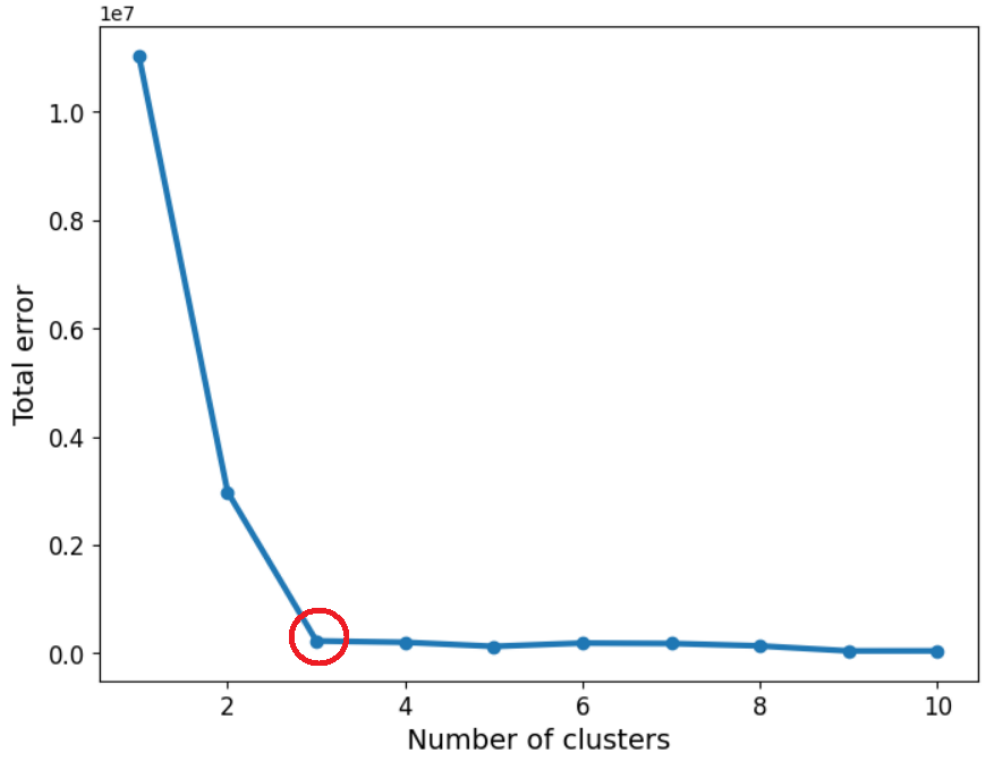


Nhận thấy 3 điểm trung tâm đã thay đổi và phân cụm rõ

```
err_total =[]
n=10
df_elbow = blods[['x','y']]
for i in range(n):
    _,my_errs,_ = kmeans(df_elbow,i+1)
    err_total.append(sum(my_errs))
```

```
fig, ax = plt.subplots(figsize=(8, 6))
plt.plot(range(1,n+1),err_total,linewidth=3,marker='o')
ax.set_xlabel(r'Number of clusters', fontsize=14)
ax.set_ylabel(r'Total error', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```

Dùng elbow để xem giá trị k chọn có hợp lý hay không



Nhận thấy tại vị trí thứ 3 đồ thị xấp xỉ không còn thay đổi
=> giá trị k là hợp lý